L Number	Hits	Search Text	DB	Time stamp
-	550582	(detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:22
-	22633	(repair$4 or fix$4 or heal$4) adj3 (program$ or list$ or function$ or subroutine$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:30
-	155	((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$ or function$ or subroutine$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:22
-	221134	(detect$4 or track$4 or monitor$4 or check$4) adj5 (fail$4 or error$4 or problem$ or fault$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:23
-	23958	(repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:44
-	67	((repair$4 or fix$4 or heal$4) adj3 (program$ or list$ or function$ or subroutine$)) with ((detect$4 or track$4 or monitor$4 or check$4) adj5 (fail$4 or error$4 or problem$ or fault$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:24
-	8	(((repair$4 or fix$4 or heal$4) adj3 (program$ or list$ or function$ or subroutine$)) with ((detect$4 or track$4 or monitor$4 or check$4) adj5 (fail$4 or error$4 or problem$ or fault$4))) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:29
-	556967	remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:29
-	8215	(repair$4 or fix$4 or heal$4) adj3 (program$ or list$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:58

-	0	((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) with remote with ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:36
-	0	((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) same ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) same remote same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:36
-	2376	((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:36
-	1	((((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))) and remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:37
-	11	(((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:37
-	1		USPAT	2003/08/07 14:41
-	1		USPAT	2003/08/07 14:41
-	515	(repair$4 or fix$4 or heal$4) adj3 malfunction	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:49
-	24412	((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:50
-	2382	(((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:50

-	11	((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:50
-	0	(((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))) not ((((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:51
-	18	((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) and ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$)) and remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:51
-	17	(((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) and ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$)) and remote) not (((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:55
-	3370	(714/?).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:55
-	5007	(717/1??).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:55

-	0	(717/2??).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:55
-	0	(717/3??).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:56
-	0	(717/?).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:57
-	8352	((714/?).ccls.) or ((717/1??).ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:57
-	16062	(repair$4 or fix$4 or heal$4) adj3 (program$ or application$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:59
-	121	((repair$4 or fix$4 or heal$4) adj3 (program$ or application$)) with (((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 15:01
-	3	(((repair$4 or fix$4 or heal$4) adj3 (program$ or application$)) with (((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction))) same remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 15:03
-	7	((((repair$4 or fix$4 or heal$4) adj3 (program$ or application$)) with (((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction))) and remote) and (((714/?).ccls.) or ((717/1??).ccls.))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 15:03
-	34	(((repair$4 or fix$4 or heal$4) adj3 (program$ or application$)) with (((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction))) and remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/09 20:22
-	2	("5793497").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/09 20:48

-	2	("5875308").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/09 20:49

US-PAT-NO: 5655069

DOCUMENT-IDENTIFIER: US 5655069 A

TITLE: Apparatus having a plurality of programmable logic
processing units for self-repair

---------- KWIC ---------

Detailed Description Text - DETX (214):
In the information processing apparatus with self**-repair function according to the eighth embodiment of the present invention, the fault** detecting circuit 21 shown in FIG. 26, for example, **detects that a fault** occurs in the logic cell 36 of the logic module 133, the reconfiguration data computing means 35 calculates reconfiguration data, based on data in the reconfiguration data holding mechanism 22, using as a spare logic cell the logic cell 136 in the logic module 134, instead of a logic cell in the logic module 133.

US-PAT-NO: 5317573

DOCUMENT-IDENTIFIER: US 5317573 A

TITLE: Apparatus and method for real time data error capture and compression redundancy analysis

---------- KWIC ---------

Detailed Description Text - DETX (15):

Second, as a rule, a row or column is designated as a "must fix" row or column as soon as the number of failures detected along that row or column reaches (hereinafter, "must fix failure number") the number of the opposite columns or rows, respectively, which is available for repair. The rationale for this rule is that it is more economical, in terms of testing time and in terms of use of redundancy, immediately to designate and use a single redundancy row or column to **repair the detected failures,** when it is known that to make repairs using the opposite columns or rows, respectively, would take additional **repair (programming**) time and exhaust all the opposite columns or rows, respectively, to **repair the same detected failures**. In the present example, there are four redundancy rows 110A and four redundancy columns 120A which can be used for repairing the memory array (A). Accordingly, as illustrated in FIG. 2, each of the columns 1-4 assigned and associated with the addresses ADDR X.sub.1, ADDR X.sub.2, ADDR X.sub.3 and ADDR X.sub.4, respectively, has an error count of "4", and is therefore designated as a "must fix" column (indicated by the logical value "1" along the MUST FIX entries). Similarly, each of the rows 17-20 assigned and associated with the addresses ADDR Y.sub.17, ADDR Y.sub.18, ADDR Y.sub.19 and ADDR Y.sub.20, respectively, has an error count of "4", and is therefore designated as a "must fix" column (indicated by the logical value "1" along the MUST FIX entries).

for inputting data through touching of selected icons displa

DERWENT-ACC-NO: 1996-198031

DERWENT-WEEK: 199620

TITLE: Data processor fault monitoring e.g. generator frequency - by suspending fixed fault monitoring function when included in new maintenance system

PATENT-ASSIGNEE: NIPPON DENKI ENG KK[NIDE]

PRIORITY-DATA: 1994JP-0204808 (August 30, 1994)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
JP 08069388 A	March 12, 1996	N/A	005	G06F 011/20

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO	APPL-DATE
JP 08069388A	N/A	1994JP-0204808	August 30, 1994

INT-CL (IPC): G06F011/20, G06F011/34

ABSTRACTED-PUB-NO: JP 08069388A

BASIC-ABSTRACT:

The method involves reporting a generated fault to a diagnostic controller (2). A data-processor (1) has a power-supply switch on report function (1-2) for reporting the power supply activation. A fault processing function (2-1) receives the fault report of the data-processor. A **fixed fault monitoring function (2-3) fixes the fault** time monitoring of the data-processor. A fixed fault monitoring stop function (2-5) suspends the fixed fault monitoring function.

The suspension occurs when the power supply switch-on report is received from the data processor. The power supply switch on the detection function (2-4) starts the fixed fault monitoring stop function. The power supply information area and the monitoring time information area displays a present power

supply
state and the present monitoring time state respectively.

ADVANTAGE - Recognises fixed fault from intermittent fault; improves data-processing system reliability.

CHOSEN-DRAWING: Dwg.1/2

DERWENT-CLASS: T01

EPI-CODES: T01-G03; T01-G05C;

US-PAT-NO: 5333308

DOCUMENT-IDENTIFIER: US 5333308 A

TITLE: Method and apparatus for operating a communication network monitor arrangement

---------- KWIC ---------

Detailed Description Text - DETX (25):

REPAIR PROGRAM: This is the program that **repairs if an error or a problem is detected** by the status program. If there is no **repair program,** then this field will be null. All the required options for this **repair program** are also specified.

TDB-ACC-NO: NNRD410119

DISCLOSURE TITLE: System Programmable Logic Failsafe Mechanism

PUBLICATION-DATA: Research Disclosure, June 1998, UK

VOLUME NUMBER: 41

ISSUE NUMBER: 410

PUBLICATION-DATE: June 1, 1998 (19980601)

CROSS REFERENCE: 0374-4353-41-410-0

DISCLOSURE TEXT:

This document contains drawings, formulas, and/or symbols that will not appear on line. Request hardcopy from ITIRC for complete article.

A failsafe method for ensuring all system programmable devices are programmed prior to product ship is disclosed. If an escape in the manufacturing process fails to program one or more devices, the system will not boot, and a signal indicating the problem will be available.

Most of the digital circuits in a modern computer system are non-programmable (i.e. unchangeable) circuits. These circuits are inside chips known as ASICs or Gate Arrays.

However, there generally is some percentage of digital circuits that reside in "programmable devices", such as PALs, GALs, and Field Programmable Gate Arrays (FPGA's). Programmable devices are blank until they are programmed, and are often programmed with custom digital circuits. Programmable devices vary in many ways such as the size of the design they can hold and the speed with which inputs affect outputs or internal states.

Programmable devices are more expensive than their non-programmable counterparts, but are still used in certain instances where flexibility is important.

Some of the main uses for programmable devices are: 1) a design where the requirements are not firm and is likely to change during development -- the greatest feature of programmable logic is that the device can easily be reprogrammed with a different design should requirements be changed or problems with the initial design be found. 2) As specialized glue logic: While most of the system can be realized with off-the-shelf chips, there is a requirement for some specialized glue logic that does not exist in any of the off the shelf parts -- this specialized logic can be put in a programmable device rather than go through the large effort and expense of making a real ASIC just for the extra glue logic. 3) Bug fixes: Very often during development, bugs are found in one or more of the main system chips.

Many times these chips will not be "turned" in time to fix those bugs so the product must ship with the chips as they are. A fix can often be placed in a programmable device that can interface to the broken chip to fix or avoid the problem.

Since programmable devices start out blank, they must be programmed as part of the system or board build process. Manufacturing processes exist for programming each device with the correct code. Most of the time, these manufacturing processes work fine. But the authors of this disclosure have noticed that inevitably there will be escapes from time to time. One or more of the programmable devices will inadvertently be left unprogrammed, i.e. blank.

Most of the time Manufacturing can catch this problem before ship time because in most cases the system will fail their ship criteria tests due to the device being blank. However, in some cases, the tests that Manufacturing performs prior to ship may not reveal the problem because 1) not every feature of the machine is tested and 2) bugs **fixed by programmable** devices are rarely encountered in manufacturing tests because the devices are fixing some intermittent problem. Shipping systems with such parts blank leaves those problems lurking and, under the right circumstances, the customer will experience the problem.

This is especially dangerous when the programmable device is intended to fix a problem that causes intermittent data corruption in memory or an I/O device. These types of fixes are not too uncommon, but it may be difficult to **detect** the absence of the **fix since the problem** being fixed is intermittent.

The ideal solution to this problem is to keep improving the manufacturing process until this doesn't happen. However, we've noticed that no matter how much Manufacturing tries to prevent this from happening, given the right set of circumstances, this will eventually occur in some systems. We believe the only failsafe solution is to create a situation in which the machine is rendered disabled until all programmable devices are programmed.

Essentially a signal called ALL_PROGRAMMED is gated into the system reset logic and will hold the system in reset unless all the programmable devices are programmed. This ensures that the system cannot escape the ship level tests without detecting the problem. If a machine under test does not boot up, the tester can probe the ALL_PROGRAMMED signal to see if the problem is an unprogrammed module, or an LED indicator can be present to indicate ALL_PROGRAMMED, although using an LED would incur additional expense.

There are two ways of creating the ALL_PROGRAMMED signal; one may work better than the other depending on the allocation of free pins on programmable devices.

The first method, shown in

figure 1, uses one pin called PROGRAMMED on each programmable device to indicate that the device is programmed. A weak pulldown resistor is placed on the PROGRAMMED signal of each device which ensures that PROGRAMMED will read '0' when the device is blank since all unprogrammed devices tri-state their I/O's. Part of the design in the programmable device sets PROGRAMMED <= '1' so the PROGRAMMED signal will read '1' when the device is programmed. Each programmable device outputs such a signal and all these signals are fed to the programmable device that has the greatest number of free I/O. This device generates the ALL_PROGRAMMED signal by internally ANDING all the individual PROGRAMMED signals together. Inside this device is the following circuit: SEE ORIGINAL.